7/12/2005

A “Vector” Model of Trust

A new model to assess multiple levels of trust

There are currently two models of trust – the binary models and the non-binary models. The binary models assign a value to trust of either 0 (no trust) or 1 (complete trust). The non-binary models assign quantitative values in the range 0 to 1 or qualitative values like high, medium, or low. The values of these models tend to be subjective estimates. Both these models have significant shortcomings that render them inadequate for many decision-making processes, and are therefore of very little help to the decision maker who must often make quick decisions based on the trustworthiness of information he/she has received. The models have no accepted formalism for the specification of trust or any method to address the dependence of trust on time. There are no methods for measuring trust, comparing trust values, and composing trust values.

We have developed a new model that facilitates answering questions about complex composed systems such as:

1. For what mission critical activities can the system be trusted?
2. In what circumstances can I trust the system?

The “Vector” Model of Trust

The “Vector” Model of Trust provides an objective decision support system, determines trust values, facilitates comparing trust values of two systems, negotiates and manages trust values, and computes the trust value of composed systems (given the trust values of the component subsystems). The model provides a gap analysis tool to facilitate the development of more trustworthy systems.

Model Elements

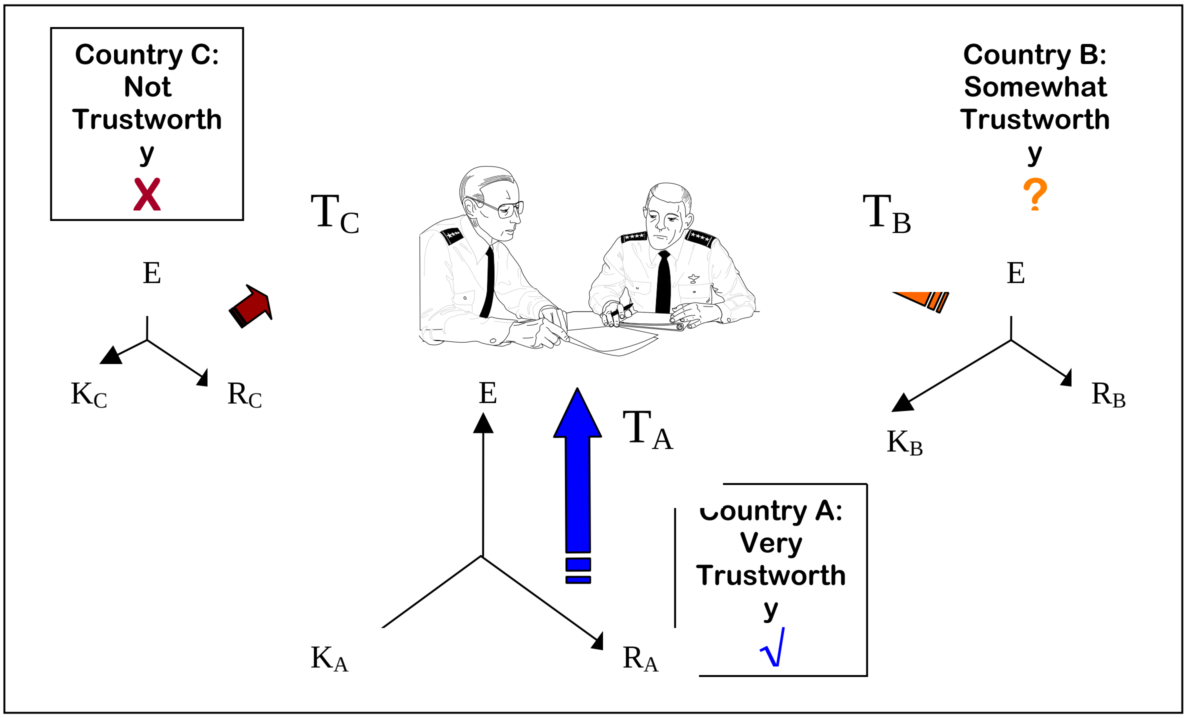
1. A trust (distrust) relationship exists between a truster and a trustee as a three-element vector of numeric values:
    - Experience – E.g. history
    - Knowledge – E.g. specifications
    - Recommendation – E.g. personal reports
2. Trust comparison and composition are accomplished using the three-element vector of numeric values in 1. above.
3. Trust dynamics
    - Trust depends on previously established trust values and can change with time if not updated.
4. A trust management system to formulate, store and manage trust vectors.

System Requirements

1. External interface for interacting with the physical world.
2. An SQL type language (TrustQL) to query and manipulate trust relationships.
3. Databases to store and manage information about trust relationships.
4. Trust monitor to provide quantitative feedback in trust negotiation.

Scenarios

1. Scenario 1
   - Access to restricted communications system XYZ does not necessarily define the level of trust of the users to each other.
     - Integrity, confidentiality, authentication, authorization, etc. are difficult to ascertain in complex environments where insufficient information about entities is available or where traditional cryptographic techniques are not appropriate.
   - Vector model of trust provides soft security to answer questions like who to trust, how much, and why. This is in contrast to hard security such as a rules based firewall.
2. Scenario 2
   - A trusts B and B trusts C. How much can A trust C?
3. Scenario 3
   - A trusts B some amount *x* to provide rain and snow forecasts. A trusts C some amount *y* to provide hurricane and tornado forecasts. How much can A trust the composed system of B and C to provide weather forecasts? (graphical interpretation below)



On-going and Future Tasks

1. Refine the TrustQL language.
2. Develop a formal framework for expressing and reasoning about trust context and trust chains.
3. Verification and validation of the model in different environments.
4. Provide security and privacy trust values in real-time.
5. Develop and prototype user-friendly software application of this new model of trust.